# Sprawozdania Wojciech Wnuk IO 7.15

## Zaawansowane programowanie w Javie VII semestr

Rok akademicki: 2023/24

#### LABORATORIUM 1. STRUMIENIE I PLIKI

Zadanie 1.1. Operacje na strumieniach

```
🧿 Main.java
      package com.company;
      import java.io.*;
       import java.util.*;
       import java.util.stream.*;
       public class Main {
           public static void main(String[] args) {
               // Zadanie 1.1: Operacje na strumieniach
               List<String> przedmioty = new ArrayList<>();
               przedmioty.addAll(Arrays.asList("Integracja systemow", "Interakcja człowiek-komputer",
               Stream<String> strumienPrzedmiotow = przedmioty.stream();
               List<Integer> listaOcen = strumienPrzedmiotow
                       .filter(przedmiot -> !przedmiot.contains("Zaaw"))
                       .map(przedmiot -> {
                           return (int) (Math.random() * 4) + 2;
                       })
                       .collect(Collectors.toList());
               listaOcen.forEach(ocena -> System.out.println("1.1: Ocena: " + ocena));
               Map<Integer, Long> liczbaPowtorzen = listaOcen.stream()
                       .collect(Collectors.groupingBy(Integer::intValue, Collectors.counting()));
               liczbaPowtorzen.forEach((ocena, liczba) -> {
                       System.out.println("Ocena " + ocena + " powtarza sie " + liczba + " razy.");
              });
```

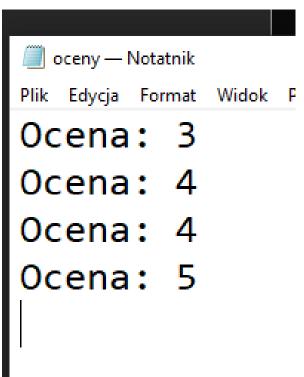
#### Zadanie 1.2. Operacje na plikach

#### Zadanie 1.3. Operacje na plikach

```
// Zadanie 1.3: Operacje na plikach
List<String> ocenyIZPliku = new ArrayList<>();
    BufferedReader reader = new BufferedReader(new FileReader( fileName: "oceny.txt"));
   String linia;
    while ((linia = reader.readLine()) != null) {
        ocenyIZPliku.add(linia);
    reader.close();
} catch (IOException e) {
    e.printStackTrace();
double srednia = ocenyIZPliku.stream() Stream<String>
        .mapToDouble(ocena -> Integer.parseInt(ocena.substring(7))) DoubleStream
        .average() OptionalDouble
        .orElse( other: 0);
Optional<String> najlepszaOcena = ocenyIZPliku.stream()
        .max(Comparator.comparingInt(ocena -> Integer.parseInt(ocena.substring(7))));
Optional<String> najgorszaOcena = ocenyIZPliku.stream()
        .min(Comparator.comparingInt(ocena -> Integer.parseInt(ocena.substring(7))));
System.out.println("Średnia ocen: " + srednia);
System.out.println("Najlepsza ocena: " + najlepszaOcena.orElse( other: "Brak ocen"));
System.out.println("Najgorsza ocena: " + najgorszaOcena.orElse( other: "Brak ocen"));
```

#### Wyniki działania kodu:

```
"C:\Program Files\Java\jdk-17\bin
1.1: Ocena: 4
1.1: Ocena: 3
1.1: Ocena: 5
1.1: Ocena: 4
□ Ocena 4 powtarza się 2 razy.
□ Średnia ocen: 4.0
Najlepsza ocena: Ocena: 5
Najgorsza ocena: Ocena: 3
```



#### Wnioski:

Wykorzystanie strumieni w Javie pozwala na wygodne i efektywne przetwarzanie danych, zarówno w przypadku operacji na kolekcjach, jak i operacji na plikach. Funkcje takie jak filter, map i collect są użytecznymi narzędziami do manipulacji danymi w strumieniach, umożliwiając filtrowanie, transformację i zbieranie wyników w odpowiednich strukturach danych.

### LABORATORIUM 2. WYKORZYSTANIE PLIKÓW XML.